

Page 15, line 32, replace "be come" with --become--.

Page 20, line 22, replace "resister" with --register--.

Page 26, line 3, after "dependent" (before the comma) insert --situations--.

***In the Claims:***

Please cancel claim 1 without prejudice or disclaimer.

Please add the following new claims:

*Sub C1* 12. A system for register renaming in a computer system capable of out-of-order instruction execution, comprising:

*B1* a temporary buffer comprising a plurality of storage locations for storing execution results, wherein an execution result for an instruction in an instruction window is stored at one of said plurality of storage locations, said one of said plurality of storage locations being assigned to said instruction in said instruction window; and

tag assignment logic for receiving data dependency results from a data dependency checker and for outputting a tag comprising an operand address for an operand of a first instruction, wherein said operand address is an address of said operand in one of said plurality of storage locations if said first instruction is dependent on a previous one of said plurality of instructions in said instruction window for said operand.

*2*  
*13* 13. The register renaming system of claim *12*, further comprising termination logic that transfers the execution results in said plurality of storage locations in said

33

B

temporary buffer to register file locations in-order based on the order of instructions in said instruction window.

<sup>3</sup>/~~14~~. The register renaming system of claim <sup>2</sup>/~~3~~, wherein said termination logic transfers a plurality of execution results from said temporary buffer to said register file simultaneously.

<sup>4</sup>/~~15~~. The register renaming system of claim <sup>3</sup>/~~14~~, wherein said termination logic transfers an execution result for an instruction from said temporary buffer to said register file when all execution results for all prior instructions are retireable.

<sup>5</sup>/~~16~~. The register renaming system of claim <sup>1</sup>/~~2~~, wherein said tag further comprises a 1-bit identifier that indicates whether said address within said tag is an address within a register file or within said plurality of storage locations.

<sup>6</sup>/~~17~~. The register renaming system of claim <sup>1</sup>/~~2~~, further comprising register file port MUXes that pass said tags to read address ports of said temporary buffer for accessing said instruction execution results.

<sup>7</sup>/~~18~~. A computer system, comprising:  
a memory unit for storing program instructions;  
a bus coupled to said memory unit for retrieving said program instructions; and

34

B

a processor coupled to said bus, wherein said processor comprises a register renaming system, comprising:

a temporary buffer comprising a plurality of storage locations for storing execution results, wherein an execution result for an instruction in an instruction window is stored at one of said plurality of storage locations, said one of said plurality of storage locations being assigned to said instruction in said instruction window; and

tag assignment logic that receives data dependency results from a data dependency checker and outputs a temporary buffer storage location address in place of a register address for an operand of a first instruction if said first instruction is dependent on a previous one of said plurality of instructions in said instruction window for said operand, wherein said temporary buffer storage location address is an address of said operand in one of said plurality of storage locations.

<sup>8</sup>  
~~19.~~ The computer system of claim <sup>7</sup>~~18~~, wherein said processor further comprises termination logic that transfers the execution results in said plurality of storage locations in said temporary buffer to register file locations in-order based on the order of instructions in said instruction window.

<sup>9</sup>  
~~20.~~ The computer system of claim <sup>8</sup>~~19~~, wherein said termination logic transfers a plurality of execution results from said temporary buffer to said register file simultaneously.

<sup>10</sup>  
~~21.~~ The computer system of claim ~~20~~<sup>9</sup>, wherein said termination logic transfers an execution result for an instruction from said temporary buffer to said register file when all execution results for all prior instructions are retirable.

<sup>11</sup>  
~~22.~~ The computer system of claim ~~18~~<sup>7</sup>, wherein said tag comprises an address and a 1-bit identifier that indicates whether said address within said tag is an address within a register file or said plurality of storage locations.

<sup>12</sup>  
~~23.~~ The computer system of claim ~~18~~<sup>7</sup>, wherein said processor further comprises register file port MUXes that pass said tag to read address ports of said temporary buffer for accessing said execution results.

<sup>13</sup>  
~~24.~~ A register renaming method, comprising the steps of:

- (1) storing, in a temporary buffer, out-of-order execution results in storage locations assigned to instructions in an instruction window;
- (2) generating at least one tag to specify an address in said temporary buffer at which said out-of-order execution results are temporarily stored; and
- (3) outputting one of said at least one tag comprising an address in place of a register address for an operand of a first instruction if a data dependency result indicates that said first instruction is dependent on a previous instruction in said instruction window, wherein said tag comprises an address of said operand in said temporary buffer.

20

6

<sup>14</sup>/~~25~~. The register renaming method of claim <sup>13</sup>/~~24~~, further comprising the step of transferring said out-of-order execution results in said temporary buffer to a register file in-order based on the order of instructions in said instruction window.

<sup>15</sup>/~~26~~. The register renaming method of claim <sup>14</sup>/~~25~~, further comprising the step of transferring a plurality of execution results from said temporary buffer to said register file simultaneously.

<sup>16</sup>/~~27~~. The register renaming method of claim <sup>15</sup>/~~26~~, further comprising the step of transferring an out-of-order execution result from said temporary buffer to said register file when all execution results for all prior instructions are retireable.

<sup>17</sup>/~~28~~. The register renaming method of claim <sup>13</sup>/~~24~~, further comprising the step of determining data dependencies between the instructions in said instruction window to produce said data dependency results.

<sup>18</sup>/~~29~~. The register renaming method of claim <sup>13</sup>/~~24~~, wherein said step (2) further comprises the step of generating tags that comprise an address and a 1-bit identifier that indicates whether said address within said tags is an address within a register file or said temporary buffer.